

**System and Method for Estimating Service Oriented  
Labor Costs**

**BACKGROUND OF THE INVENTION**

**1. Technical Field**

5       The present invention relates in general to a system  
and method for estimating service oriented labor costs.  
More particularly, the present invention relates to a  
system and method for providing detailed labor costs  
corresponding to an urgency level in order to more  
10 accurately plan, measure, and manage a labor pool.

**2. Description of the Related Art**

Companies experience unprecedented pressure to provide  
quality service at reduced prices. Companies continuously  
search for ways to achieve these two seemingly  
15 contradicting goals. A company may reduce prices too much  
which decreases profit if operating costs are not reduced  
accordingly. Some companies reduce headcount in order to  
maintain profit margins. However, headcount reduction many  
times decreases quality of service to customers.

20       Companies are encouraged to have a clear understanding  
of labor costs in order to effectively respond to customer  
requests. When a company understands the cost of  
performing a particular customer request, the company may  
successfully bid on the request and know the profit gained  
25 if the company wins the business.

Companies may choose to "low ball" a customer request  
in order to win business. For example, a new company

20044189-010902

attempting to enter a marketplace may bid on customer requests at "cost", or without making a profit in order to be the lowest bidder and win the business.

5 In order to effectively "low ball" a request, the new company should understand its operating costs. Otherwise, the company may under bid a customer request and lose money. A challenge found is determining detailed labor rates in order to understand actual operating costs.

10 Labor rates are typically categorized by skill levels. For example, a senior technician may have one standard labor rate, while a junior technician may have a different standard labor rate. Standard labor rates, however, lack detail to accurately determine labor costs for bidding on a particular level of service, or level of urgency.

15 The urgency of a customer request directly affects the actual cost of responding to the customer request. For example, if a customer requires a service completed within one business day, a senior technician may be delegated to that request and may work overtime in order to finish the service within one business day.

20 On the other hand, when a customer requires a service completed within one week, the senior technician may work on multiple customer requests reducing his unapplied hours and may not work overtime. A challenge found is accurately tracking labor costs corresponding to a customer requested service level.

25 What is needed, therefore, is a way to accurately bid on customer requests that fluctuate in the level of service.

**SUMMARY**

It has been discovered that accurate bidding on customer requests is achieved by generating labor indices by service level and applying them to standard labor rates.

- 5 The result is multiple labor rates by service level that accurately accounts for various customer levels of urgency.

- 10 A customer request is received and resource types are determined to perform the corresponding service. Assuming that the resource types are available, corresponding labor indices by service levels are calculated and applied to standard labor rates which results in labor rates by service levels. Profit and non-labor costs are added, and a bid is sent to the customer.

- 15 Labor indices by service level are calculated using two fundamental inputs which are utilization indices and overtime indices. Utilization indices correspond to the utilization of a service level for a particular platform. Overtime indices correspond to the increase or decrease in the amount of overtime for a particular service level.

- 20 Utilization indices by service level are calculated using two primary inputs which are utilization weighting and utilization improvement. Utilization weighting is an averaging factor for determining utilization indices for each service level. Utilization improvement is an increase  
25 in applied/billable hours. For example, a business may spend 100 hours on a project, but the business is only able to bill 60 hours (60% utilization). If they are to bill 65 hours (65% utilization) using a particular service level, the utilization improvement is 5%.

205010" 6442400F

Overtime indices by service level are calculated using two primary inputs which are max rate mix plan and overtime labor factor. Max rate mix plan is an averaging factor based on estimated overtime indices for each service level.

- 5 Overtime labor factor corresponds to the increase or decrease in the cost of overtime for a particular service level.

- 10 Utilization indices by service level and overtime indices by service level are principal factors in generating labor indices by service level. Labor indices by service level are then applied to standard labor rates to generate labor rates by service level which are used to generate customer bids.

- 15 The foregoing is a summary and thus contains, by necessity, simplifications, generalizations, and omissions of detail; consequently, those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects, inventive features, and advantages of the present  
20 invention, as defined solely by the claims, will become apparent in the non-limiting detailed description set forth below.

206070-634400T

**BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings. The use of the same reference symbols in different drawings indicates similar or identical items.

**Figure 1** is a flowchart showing steps taken in processing a customer request and responding to the request;

**Figure 2** is a high-level diagram showing key inputs generating multiple labor rates by service level;

**Figure 3** is a flowchart showing steps taken in calculating labor rates by service level corresponding to a platform;

**Figure 4** is a flowchart showing steps taken in calculating utilization indices corresponding to service levels;

**Figure 5** is a flowchart showing steps taken in calculating overtime indices corresponding to service levels;

**Figure 6** is a flowchart showing steps taken in calculating an overtime labor factor; and

**Figure 7** is a block diagram of an information handling system capable of implementing the present invention.

**DETAILED DESCRIPTION**

The following is intended to provide a detailed description of an example of the invention and should not be taken to be limiting of the invention itself. Rather,  
5 any number of variations may fall within the scope of the invention which is defined in the claims following the description.

**Figure 1** is a flowchart showing steps taken in processing a customer request and responding to the  
10 request. Estimator processing commences at **100**, whereupon customer request **110** is received and analyzed (step **105**). Customer request **110** may be a request to manufacture a product or provide a service in particular timeframe. For example, a customer may request a service completed by the  
15 next business day.

Resources adjusted for a service level corresponding to the customer request are identified in request resource needs **120** (step **115**). Request resource needs **120** may be stored in a non-volatile storage area, such as a computer  
20 hard drive. The first resource needed is retrieved from request resource needs **120** at step **125**, and its availability is retrieved from organization resources **135** at step **130**. Organization resources **135** includes the availability of resources in an organization and may be  
25 stored in a non-volatile storage area, such as a computer hard drive.

A determination is made as to the resource availability during corresponding timeframe of the customer request (decision **145**). If the resource is not available,

decision **145** branches to "No" branch **146** whereupon a bid is not generated (step **150**), and processing ends at **155**. For example, resources may be identified in the customer request that are preoccupied with other requests.

5        In another embodiment, a bid may be generated with available resources that may not completely match the customer request. However, the bid may be lower due to inconveniencing the customer. For example, the customer may request a product to be delivered overnight. For  
10 various reasons, resources to deliver the product overnight may not be available. Processing may determine the most comparable resource available that delivers the product in two business days and send a corresponding bid to the customer.

15        On the other hand, if resources are available for the timeframe corresponding to the customer request, decision **145** branches to "Yes" branch **148** whereupon labor rates are computed by service level (pre-defined process block **160**, see **Figure 3** for further details). A determination is made  
20 as to whether a standard labor rate is higher than the computed estimate labor rate by service level (decision **165**). For example, processing may compute an estimate labor rate by service level lower than the standard labor rate. In order to gain more revenue and profit, processing  
25 may choose the higher standard labor rate for bidding purposes.

      If the standard labor rate is more than the estimate labor rate by service level, decision **165** branches to "No" branch **166** whereupon the standard labor rate is stored in  
30 labor rates for request **175** (step **170**). Labor rates for

200449084001

request **175** may be stored in a non-volatile storage area, such as a computer hard drive.

On the other hand, if the estimate labor rate by service level is higher than the standard labor rate, decision **165** branches to "No" branch **168** whereupon The estimate labor rate by service level is stored in labor rates for request **175** (step **180**).

A determination is made as to whether there are more resources required to generate a customer bid corresponding to the customer request (decision **185**). If more resources are required, decision **185** branches to "Yes" branch **186** which loops back to read (step **140**) and process the next resource requirement. This looping continues until there are no more resource requirements, at which point, decision **185** branches to "No" branch **188**. A labor and non-labor component bid is computed based on labor rates for request **175** (step **190**). Profit is added and the bid is sent to the customer at step **195**. Processing ends at **199**.

**Figure 2** is a high-level diagram showing fundamental inputs generating multiple labor rates by service level. Labor indices by service level **200** generates multiple labor indices that correspond to a particular service level. For example, one level of service is when a customer requests a product or service on the same day of his request. Another level of service is when a customer requests a product or service within one week of his request.

Multiple labor indices by service levels **290** are multiplied with single labor rate **270** to obtain multiple labor rates by service levels **280**. Single labor rate **270** may be a current labor rate based on skill level, such as



the hourly cost of a senior technician. Using the example above, the labor rate for a technician when a customer requests same day service may be higher than the labor rate for a technician when a customer requests service within  
5 one week.

Labor indices by service levels **200** are computed using two primary inputs. The two primary inputs are utilization indices by service levels **210** and overtime indices by service levels **240**. Utilization indices by service levels  
10 **210** correspond to the utilization of a service level for a particular platform. Overtime indices by service levels **240** correspond to the increase or decrease in the amount of overtime for a particular service level

Utilization indices by service levels **210** are  
15 calculated using two primary inputs which are utilization weighting **220** and utilization improvement by service level **230**. Utilization weighting **220** is an averaging factor for determining utilization indices for each service level. Utilization improvement by service level **230** corresponds to  
20 an increase in applied/billable hours.

Overtime indices by service level **240** are calculated using two primary inputs which are max rate mix plan **260** and overtime labor factor by service level **250**. Max rate mix plan **260** is an averaging factor for determining  
25 overtime indices for each service level. Overtime labor factor by service level corresponds to the increase or decrease in the amount of overtime for a particular service level.

**Figure 3** is a flowchart showing steps taken in  
30 calculating labor rates by service level corresponding to a

platform. Processing commences at 300, whereupon platform information is retrieved from platform store 315 (step 310). For example, platform information may include the labor requirements to build a particular product.

5 Utilization indices are generated and stored in utilization output store 325 (pre-defined process block 320, see **Figure 4** for further details). Overtime indices are generated and stored in overtime output store 335 (pre-defined process block 330, see **Figure 5** for further  
10 details).

Labor indices by service level are calculated and stored in labor index store 345 (step 340). The calculation uses utilization indices (UI) from utilization output store 325 and overtime indices (OI) from overtime  
15 output store 335. In one embodiment, labor indices by service level (LISL) are calculated using the following formula:

$$\text{LISL} = 1 + (\text{UI}-1) + (\text{OI}-1)$$

However, other formulas may be used which result in a  
20 similar labor index by service level.

Standard labor rates are retrieved from standard labor rate store 355 (step 350). Labor rates by service level are calculated and stored in LRSL store 365 at step 360. Labor rates by service level (LRSL) are calculated using  
25 standard labor rates (SLR) and labor indices by service level (LISL). In one embodiment, labor rates by service level are calculated using the following formula:

$$\text{LRSL} = \text{SLR} * \text{LISL}$$

However, other formulas may be used which result in a similar labor rate by service level.

A determination is made as to whether there are more standard labor rates (decision **370**). If there are more  
5 standard labor rates, decision **370** branches to "Yes" branch **372** which loops back to retrieve and process the next standard labor rate. This looping continues until there are no more standard labor rates to process, at which point decision **370** branches to "No" branch **378**.

10 A determination is made as to whether there are more platforms to process corresponding to the customer request (decision **380**). If there are more platforms to process, decision **380** branches to "Yes" branch **382** which loops back to process the next platform. This looping continues until  
15 there are no more platforms to process, at which point decision **380** branches to "No" branch **388**. Processing ends at **390**.

**Figure 4** is a flowchart showing steps taken in calculating utilization indices corresponding to service  
20 levels. Processing commences at **400**, whereupon a labor mix is retrieved from utilization input store **415** (step **410**). A labor mix corresponds to the mix of service level for a particular platform or product line.

A determination is made as to whether the labor mix is  
25 zero or not available (decision **420**). If the labor mix is zero or not available, decision **420** branches to "Yes" branch **422** whereupon "Not Available" is stored in utilization output store **455** (step **450**). On the other hand, if the labor mix is not zero, decision **420** branches  
30 to "No" branch **428** whereupon a utilization improvement is

retrieved. Utilization improvement corresponds to an increase in applied/billable hours.

A determination is made as to whether the utilization improvement is zero or not available (decision 440). If the utilization improvement is zero or not available, decision 440 branches to "yes" branch 442 whereupon "Not Available" is stored in utilization output store 455 (step 450). On the other hand, if the utilization improvement is not zero, decision 440 branches to "No" branch 448 whereupon a utilization weighting is calculated (step 470). In one embodiment, the utilization weighting (UW) uses each utilization improvement by service level (UMSL) and labor mix by service level (LMSL) and is calculated using the following formula:

$$UW = UMSL1 * LMSL1 + UMSL2 * LMSL2 + UMSLn * LMSLn$$

where 1,2,n correspond to service levels. However, other formulas may be used which result in a similar utilization weighting factor.

A utilization index by service level is calculated and stored in utilization output store 455 (step 480). The utilization index by service level uses the utilization weighting (UW), the utilization improvement by service level (UMSL), and a utilization unit factor (UUF). The UUF converts the UISL calculation whereby a 1% increase in utilization corresponds to a factor of 1 increase is UISL. In one embodiment, the utilization index by service level (UISL) is calculated using the following formula:

$$UISL = 1 + 100 * UUF * (UW - UMSL)$$

However, other formulas may be used which result in a similar utilization index by service level.

A determination is made as to whether there are more service levels to process (decision 490). If there are  
5 more service levels to process, decision 490 branches to "Yes" branch 492 which loops back to process the next service level. This looping continues until there are no more service levels to process, at which point decision 490 branches to "No" branch 498. Processing returns at 499.

10 **Figure 5** is a flowchart showing steps taken in calculating overtime indices corresponding to service levels. Processing commences at 500, whereupon first service level is retrieved from service level store 515 (step 505). Service level store 515 may be stored in a  
15 non-volatile storage area, such as a computer hard drive. An overtime labor factor by service level is calculated and stored in overtime temp store 515 (pre-defined process block 510, see **Figure 6** for further details). A determination is made as to whether the overtime labor  
20 factor is zero or not available (decision 520). If the overtime labor factor is zero or not available, decision 520 branches to "Yes" branch 522 whereupon "Not Available" is stored in overtime output store 535 corresponding to the service level (step 530).

25 On the other hand, if the overtime labor factor is not zero (i.e. available), decision 520 branches to "No" branch 528 whereupon labor mixes by service level are retrieved from overtime input store 545 (step 540). Labor mixes by service level correspond to the service level mix for a  
30 particular platform.

Overtime labor factors by service level are retrieved from overtime temp store **515** (step **550**). A max rate mix plan is calculated and stored in overtime temp store **515** (step **560**). The max rate mix plan is a weighting factor in  
5 calculating an overtime index by service level. In one embodiment, the max rate mix plan (MRMP) uses labor mixes by service level (LMSL) and overtime labor factor by service level (OLFSL) and is calculated using the following formula:

10 
$$\text{MRMP} = \text{LMSL1} * \text{OLFSL1} + \text{LMSL2} * \text{OLFSL2} + \text{LMSLn} * \text{OLFSLn}$$

where 1,2,n correspond to service levels. However, other formulas may be used which result in a similar max rate mix plan.

An overtime index by service level is calculated and  
15 stored in overtime output store **535** (step **570**). In one embodiment, the overtime index by service level (OISL) uses the overtime labor factor by service level (OLFSL) and max rate mix plan (MRMP) and is calculated using the following formula:

20 
$$\text{OISL} = \text{OLFSL} / \text{MRMP}$$

However, other formulas may be used which result in a similar overtime index by service level.

A determination is made as to whether there are more service levels to process (decision **580**). If there are  
25 more service levels to process, decision **580** branches to "Yes" branch **582** which loops back to retrieve the next service level from service level store **515** (step **585**) and process the next service level. This looping continues until there are no more service levels to process, at which

point decision 580 branches to "No" branch 588 and processing returns at 590.

Figure 6 is a flowchart showing steps taken in calculating an overtime labor factor by service level. Processing commences at 600, whereupon a labor rate mix corresponding to a platform is retrieved from utilization input store 620 (step 610). A determination is made as to whether the labor mix is zero or not available (decision 630). If the labor mix is zero or not available, decision 630 branches to "yes" branch 632 whereupon "Not Available" is stored in utilization temp store 690 (step 640).

On the other hand, if the labor mix is not zero, decision 630 branches to "No" branch 638 whereupon an overtime savings by service level is retrieved from utilization input store 620 (step 650). Overtime savings corresponds to a reduction of overtime from the maximum overtime service level. A determination is made as to whether the overtime savings by service level is zero or not available (decision 660). If the overtime savings by service level is zero or not available, decision 660 branches to "Yes" branch 662 whereupon "Not Available" is stored in utilization temp store 690 (step 640).

An average reduction factor (ARF) is calculated at step 665. In one embodiment, ARF is calculated using labor mix by service levels (LMSL) and overtime savings by service levels (OSSL) using the following formula:

$$ARF = LMSL1 * OSSL1 + LMSL2 * OSSL2 + LMSLn * OSSLn$$

where 1,2,n correspond to service levels. However, other formulas may be used that result in a similar average reduction factor.

An overtime weighting is calculated at step **670** which  
5 corresponds to an expected overtime, and uses ARF and a target max overtime (TMO). TMO corresponds to an estimated overtime for the highest response level of service. In one embodiment, the overtime weighting (OW) is calculated using the following formula:

10 
$$OW = TMO * (1 - ARF)$$

However, other formulas may be used that result in a similar overtime weighting.

An overtime labor factor by service level is calculated and stored in utilization temp store **690** (step  
15 **680**). In one embodiment, the overtime labor factor by service level (OLFSL) uses target max overtime (TMO), plan overtime (PO), overtime weighting (OW), and overtime savings by service level (OSSL) and is calculated using the following formula:

20 
$$OLFSL = 1 - (TMO * PO * OSSL / OW)$$

However, other formulas may be used that result in a similar overtime labor factor by service level. PO corresponds to the actual overtime a business is experiencing with the current market mix. Processing  
25 returns at **695**.

**Figure 7** illustrates information handling system **701** which is a simplified example of a computer system capable

200445 0001



of performing the server and client operations described herein. Computer system 701 includes processor 700 which is coupled to host bus 705. A level two (L2) cache memory 710 is also coupled to the host bus 705. Host-to-PCI bridge 715 is coupled to main memory 720, includes cache memory and main memory control functions, and provides bus control to handle transfers among PCI bus 725, processor 700, L2 cache 710, main memory 720, and host bus 705. PCI bus 725 provides an interface for a variety of devices including, for example, LAN card 730. PCI-to-ISA bridge 735 provides bus control to handle transfers between PCI bus 725 and ISA bus 740, universal serial bus (USB) functionality 745, IDE device functionality 750, power management functionality 755, and can include other functional elements not shown, such as a real-time clock (RTC), DMA control, interrupt support, and system management bus support. Peripheral devices and input/output (I/O) devices can be attached to various interfaces 760 (e.g., parallel interface 762, serial interface 764, infrared (IR) interface 766, keyboard interface 768, mouse interface 770, and fixed disk (HDD) 772) coupled to ISA bus 740. Alternatively, many I/O devices can be accommodated by a super I/O controller (not shown) attached to ISA bus 740.

BIOS 780 is coupled to ISA bus 740, and incorporates the necessary processor executable code for a variety of low-level system functions and system boot functions. BIOS 780 can be stored in any computer readable medium, including magnetic storage media, optical storage media, flash memory, random access memory, read only memory, and communications media conveying signals encoding the

instructions (e.g., signals from a network). In order to attach computer system **701** to another computer system to copy files over a network, LAN card **730** is coupled to PCI bus **725** and to PCI-to-ISA bridge **735**. Similarly, to  
5 connect computer system **701** to an ISP to connect to the Internet using a telephone line connection, modem **775** is connected to serial port **764** and PCI-to-ISA Bridge **735**.

While the computer system described in **Figure 7** is capable of executing the invention described herein, this  
10 computer system is simply one example of a computer system. Those skilled in the art will appreciate that many other computer system designs are capable of performing the invention described herein.

One of the preferred implementations of the invention  
15 is an application, namely, a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, on a hard  
20 disk drive, or in removable storage such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. Thus, the present invention may be implemented as a computer program product for use in a  
25 computer. In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware,

or in more specialized apparatus constructed to perform the required method steps.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects and, therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be understood by those with skill in the art that if a specific number of an introduced claim element is intended, such intent will be explicitly recited in the claim, and in the absence of such recitation no such limitation is present. For a non-limiting example, as an aid to understanding, the following appended claims contain usage of the introductory phrases "at least one" and "one or more" to introduce claim elements. However, the use of such phrases should not be construed to imply that the introduction of a claim element by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an"; the same holds true for the use in the claims of definite articles.